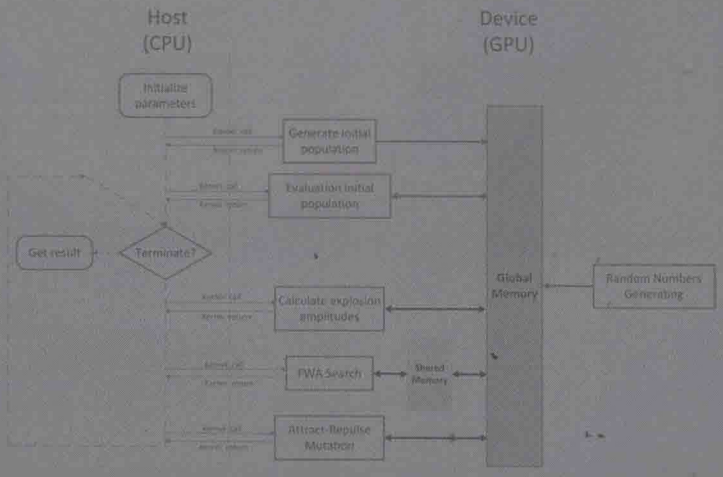
GPU-based Parallel Implementation *of* Swarm Intelligence Algorithms

Ying Tan



MORGAN KAUFMANN

GPU-based Parallel Implementation of Swarm Intelligence Algorithms

Ying Tan

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO
Morgan Kaufmann is an imprint of Elsevier

Morgan Kaufmann is an imprint of Elsevier
50 Hampshire Street, 5th Floor, Cambridge, MA 02139, USA

Notices
Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the Library of Congress

ISBN: 978-0-12-809362-7

Publisher: Todd Green
Acquisition Editor: Simon Tian
Editorial Project Manager: Naomi Robertson
Production Project Manager: Nicky Carter
Designer: Maria Inês Cruz

GPU-based Parallel Implementation of Swarm Intelligence Algorithms

To my family, Chao Deng and Fei Tan

Preface

Swarm intelligence algorithms (SIAs) are a family of intelligent optimizers inspired by the collective behavior of a swarm in nature under the umbrella of the well-known swarm intelligence (SI), which is to deal with the collective behavior of a swarm with the properties of decentralization, self-organization, and cooperation. Generally speaking, a typical SI system consists of a population of simple individuals which can communicate with each other by acting on their local environment. Though the individual in a swarm follow very simple rules, the interactions between such individuals can lead to the emergence of very complicated global behavior, far beyond the capability of single individual agent.

In SIAs, a swarm is generally made up of multiple simple individuals. The individuals can exchange heuristic information through direct or indirect local interaction. It is such interaction that, in addition to certain stochastic elements, generates the behavior of adaptive and efficient search, and finally leads to global optimization for a very complicated problem.

The two most popular SIAs are particle swarm optimization (PSO), which is inspired by the social behavior of bird flocking or fish schooling, and ant colony optimization (ACO), which simulates the foraging behavior of ant colony. Besides, many novel SIAs with differently specific search mechanisms have also been proposed continuously and achieved great successes on a number of specific problems. They are fireworks algorithm (FWA), brain storm optimization (BSO), bacterial foraging optimization (BFO), bee algorithm (BA), fish schooling search (FSS), cuckoo search, artificial bee colony (ABC) algorithm, and firefly algorithm (FA), to name a few.

Nowadays, SIAs have become efficient approaches for solving very complicated optimization problems and an effective complement of the traditional optimization methods which are prone to get trapped into local minima and can only be used for a limited range of problems. Nevertheless, SIAs suffer from the serious drawbacks of slow convergence and high computational amount, which make it infeasible for SIAs to tackle large-scale and complex problems. Therefore, it is a critical task to accelerate SIAs greatly enough to make them applicable in more scenarios.

On the other hand, current multicore revolution encourages the community to start looking at heterogeneous solutions. Heterogeneous computing, which refers to systems that use more than one kind of processor, has entered the computing's mainstream systems while diverse heterogeneous combinations have been applied in the scientific domain. The general-purpose computing on the graphics processing unit (GPGPU) is one of the most important heterogeneous solutions.

Initially designed for addressing highly computational graphics tasks, the graphics processing unit (GPU), from its inception, has many computing cores and can provide massive parallelism with thousands of cores at a reasonable price.

In the past decade, GPU has evolved into a general-purpose computing device with many cores, and has become an indispensable weapon in the arsenal of high-performance computing (HPC). Although the attempt on leveraging GPUs' massively parallel computing power can date back to the first day of GPGPU, significant progress had not been achieved until the emergence of high-level programming platforms such as compute unified device architecture (CUDA) and open computing language (OpenCL).

Based on the local interactions within population, SIAs are naturally amenable to parallelism. SIAs' such intrinsic property makes them very suitable to run on the GPU in parallel, thus achieving a remarkable performance improvement.

In the past few years, different implementations of diverse SIAs were proposed. With the excellence in performance, GPU-powered SIAs have been applied to many complex real-world problems, such as computer vision, image processing, machine learning, data mining, parameter optimization, economy as well as many other problems. Because of the significant speedup, SIAs now can be used to solve many tasks which were previously unmanageable by the original algorithm in a reasonable time.

Nowadays, all computing systems, from mobile to supercomputers, are becoming heterogeneous, massive parallel computers for higher power efficiency and computation throughput, and GPU and its likes are everywhere you are. Therefore, GPU-based SIAs (GPU-SIA) can easily find their efficient computing platform and play a very important role in many complex and large-scale problems.

Thanks to the inherent parallelism of SIAs, it is promising to leverage the highly parallel GPU to enormously speedup SIAs. The author of the book has kept track of the state-of-the-art of both domains for more than 8 years and combines them organically. Especially, in applying GPU to accelerate SIAs, we deeply studied the application of GPU to accelerate PSO, for the first time in 2009. This work has been cited more than 130 times by Google Scholar, that is a high-cited work in this domain. Very recently, we also developed a GPU-based fireworks algorithm (GPU-FWA) which achieved a beneficial of 200+ times speedup. In the year of 2015, by making full use of the cutting-edge dynamic parallelism mechanism provided by CUDA in the latest GPU platform, an attract-repulse fireworks algorithm (AR-FWA) is proposed by introducing an efficient adaptive search mechanism (AFW Search) and a nonuniform mutation strategy for spark generation, which can be implemented on the GPU easily and efficiently. Currently, it has become a popular topic to study GPU-based SIAs, which has great impact on the solving of large-scale, complicated problems.

So I think it is the time for me to organize those works, published in a variety of journals and conferences, together, which also triggered me to write this book for researchers, engineers, and graduates with interests in parallel implementation of the novel SIAs.

This monograph specifically focuses on the implementation of SIAs on the GPU, which is a very popular and promising interdisciplinary topic of academic and industrial significance. Almost all of the content of this monograph are excerpted from the research works and academic papers published by myself and my supervised graduate students. In particular, this book begins by introducing SIAs, GPU,

and combination of SIAs and GPU, with some perspectives of mine. Then a detailed description of GPGPU is presented along with a brief review. Four parallel models for GPU-based implementations of SIAs are proposed in the following chapter with examples for each model in literature, with which SIAs can be implemented easily and conveniently. Chapter 4 presents the parallel performance metrics, i.e., rectified efficiency and speedup, for a desirable objective and fruitful comparison. Then Chapter 5 points out some necessary and important considerations in implementing SIAs when using GPUs. The following successive four chapters describe in detail the typical GPU's parallel implementations of SIAs, including GPU-PSO, GPU-FWA, AR-FWA, as well as GPU-GA, GPU-DE, and GPU-ACO briefly. The Chapter 10 is devoted to the generation of random numbers, one of the most important parts of SIAs. The widely used algorithms for generating random numbers are presented followed by the empirical study on the impact of different methods on the performance of PSO. With the excellent performance, GPU-powered SIAs have been applied to many real-world and complicated problems which are previously unmanageable by the original algorithm in reasonable time. Chapter 11 presents some real-world applications which benefit greatly from the GPU parallelism. At last, it is well known that benchmarking is a key for developing and comparing varieties of optimization algorithms, so the last chapter gives a CUDA-based real-parameter optimization benchmark, i.e., cuROB, with which the test functions with diverse properties are also included and implemented efficiently with CUDA.

This monograph, at present, is an unique book dealing with the combination of SIAs and GPU, and gives such an organic organization of plenty of related GPU-SIAs' material scattered in different journals and conferences, which can definitely attract a lot of readers who are eager to accelerate SIAs based on the cheaper and powerful GPU platform.

The book provides a particular horizon which is hardly realized from the perspective of GPGPU or SIAs solely. I believe that this book will be a bridge to connect the researchers from the separate domain and can be beneficial for both.

To give the readers a thorough knowledge of this emerging interdisciplinary research area, the book not only introduces the GPGPU in sufficient detail, but gives out the concrete considerations on the implementation of SIAs on the GPU. To make it self-contained, we introduce some popular SIAs when they are first met in the book.

This book features:

1. Concise but sufficient introduction to GPGPU is able to help layman to get familiar with this emerging computing technique.
2. Implementation details of SIAs on GPU are very useful and helpful for readers to easily utilize the techniques to accelerate their programs for a promising speedup.
3. Readers from the domain of HPC can find the relatively young research domain, i.e., SIAs, very interesting, and the HPC plays a increasing key role in the emerging area.
4. Many applications presented in the book are of great help for reader to decide whether or not SIAs or GPGPU can be used in their tasks at hand.

This book is suitable for researchers and practitioners of SIAs, who want to accelerate the execution of algorithms. However, for those who are interested in GPGPU alone can also find many techniques and tips for GPU programming.

Although much effort has been made, research of GPU-SIAs is still in its infancy and rising phase, especially good and convincible performance criteria as well as canonical paralleled schema are yet to be developed. In order to better understand what achievements have been made and how it is a useful insight on the future development, I try to collect the most important and latest works published by my leading group of CIL@PKU after a detailed literature review on this field. Hopefully, light can be shed on the trends of SIAs on GPU.

Due to my limited specialty knowledge and capability, a few errors, typos, and inadequacy may appear in the book, therefore valuable comments and suggestions are warmly welcome to ytan@pku.edu.cn.

Beijing, August 2015 *Ying Tan*

Acknowledgments

I would like to deliver my thanks to my past graduates, Mr. You Zhou and Dr. Ke Ding who conducted extensive and deep research on the project of GPU-based swarm intelligence algorithms under my guidance in the past 8 years.

I want to thank Dr. Simon Tian, Acquisition Editor of Elsevier S&T Books, for his kind coordination and help in reviewing the proposal and the manuscript of this book.

While working on the topics of this book, I was supported by the Natural Science Foundation of China (NSFC) under grant no. 61375119 and the Beijing Natural Science Foundation under grant no. 4162029, and partially supported by National Key Basic Research Development Plan (973 Plan) Project of China under grant no. 2015CB352302.

Acronyms

ACO	ant colony optimization
AFW	adaptive firework
APOD	assess, parallelize, optimize, deploy
APU	accelerated processing unit
AR	attract-repulse
AR-FWA	attract-repulse fireworks algorithm
BA	bee algorithm
BFO	bacterial foraging optimization
BSO	brain storm optimization
CA	cellular automata
CCPSO	cooperative coevolutionary particle swarm optimization
CMR	combined multiple recursive generator
CPSO	clonal particle swarm optimization
CS	cuckoo search
CUDA	compute unified device architecture
DE	differential evolution
DP	double precision
ERS	elitism random selection
FSS	fish schooling search
FWA	fireworks algorithm
GA	genetic algorithm
GFSR	generalized feedback shift register
GPGPU	general-purpose computing on the GPU
GPU	graphics processing unit
GPU-FWA	GPU-based fireworks algorithm
GPU-MOPSO	multiobjective particle swarm optimization on graphics processing unit
GPU-PSO	GPU-based particle swarm optimization
LCG	linear congruential generator
MCG	multiplicative congruential generators
MOO	multiobjective optimization
MOPSO	multiobjective particle swarm optimization
MRG	multiple recursive generator
MT	mersenne twister
OpenCL	open computing language
PDF	probability distribution function
PSO	particle swarm optimization
RE	rectified efficiency
RNG	random number generator

RWS	roulette-wheel selection
SIA	swarm intelligence algorithm
SIMT	single instruction multiple thread
SMP	streaming multiple processor
SP	streaming processor
SP	single precision
SPSO	standard particle swarm optimization
TSP	traveling salesman problem
VEGA	vector evaluated genetic algorithm
VEPSO	vector evaluated particle swarm optimization

Contents